TIME AND SPACE PARTITION PLATFORM FOR SAFE AND SECURE FLIGHT SOFTWARE.

Ángel Esquinas[1], Juan Zamorano[1], Juan A. de la Puente[1], Miguel Masmano[2], and Alfons Crespo[2]

[1] *Universidad Politécnica de Madrid (UPM), Facultad de Informática, Campus de Montegancedo, Boadilla del Monte, E28660 Madrid, Spain. Email: aesquina@datsi.fi.upm.es, jzamora@fi.upm.es, jpuente@dit.upm.es*

[2] *Universidad Politécnica de Valencia (UPV), Instituto de Automática e Informática Industrial, Camino de Vera s/n, E46022 Valencia, Spain. Email: mmasmano@ai2.upv.es, alfons@disca.upv.es*

ABSTRACT

There are a number of research and development activities that are exploring Time and Space Partition (TSP) to implement safe and secure flight software. This approach allows to execute different real-time applications with different levels of criticality in the same computer board. In order to do that, flight applications must be isolated from each other in the temporal and spatial domains. This paper presents the first results of a partitioning platform based on the Open Ravenscar Kernel (ORK+) and the XtratuM hypervisor. ORK+ is a small, reliable real-time kernel supporting the Ada Ravenscar Computational model that is central to the ASSERT development process. XtratuM supports multiple virtual machines, i.e. partitions, on a single computer and is being used in the Integrated Modular Avionics for Space study. ORK+ executes in an XtratuM partition enabling Ada applications to share the computer board with other applications.

Key words: Integrated Modular Avionics; Ravenscar Profile; Partition Kernel; ARINC 653.

1. INTRODUCTION

Upcoming spacecraft avionic systems are composed by functionally independent software components that may have different levels of criticality. Therefore, those software components must be isolated from each other in such a way that faults are contained and thus the effort of integration, verification and validation is reduced. The classic approach to provide isolation in the aeronautics domain is based on a *federated* architecture where different software components are executed in different on-board computer. However, the increasing processing power of aeronautics air-bone computers opened the way to *integrated* architectures, in which several applications are executed on a single computer board. In order to provide isolation between applications, the common approach is to provide *logical partitions* based on virtual machines. Each virtual machine has a share of processor time, memory space, and other resources, in such a way that virtual machines provide time and spatial isolation to the hosted software components. Temporal isolation prevents a partition to overrun its processor time budget, and spatial isolation prevents a partition to access memory space allocated to another partition.

In the recent years European space agencies have launched research and development activities to study and use Time and Space Partition (TSP) concepts in on-board software. The introduction of partition-based avionics is a strategy to achieve higher level of integration while maintaining the properties of a federated system. TSP enables the separation of concerns between functionally independent software components to contain and isolate faults and reduce the effort of the software integration, verification and validation process. There are others advantages such as the reduction in power and weight that make integrated architectures appealing for space.

The so-called Integrated Modular Avionics (IMA) concept [11] is successfully used in the aeronautics domain. There are several IMA platform based on partition operating systems that are in charge of providing temporal and spatial isolation between partitions. The ARINC 653 standard [3] defines an architecture and an applications program interface (API) for such an operating system or *application executive* (APEX), in ARINC terms. Temporal isolation is achieved by using a two-level scheduling scheme: a global or *partition scheduler* and local schedulers. The global scheduler allocates processor time to partitions according to a round-robin or static cyclic schedule, in such a way that partitions execute in turn for the duration of a fixed slice of time (see figure 1). The local schedulers allocate processor time to partition tasks based on their priority. Spatial isolation between partitions is provided by implementing a separate address space for each partition, in a similar way as process address spaces are protected from each other in conventional operating systems.

There are multiple industrial ARINC 653 implementa-

This work has been partly funded by the Spanish Ministry of Science, project TIN2008-06766-C03 (RT-MODEL).

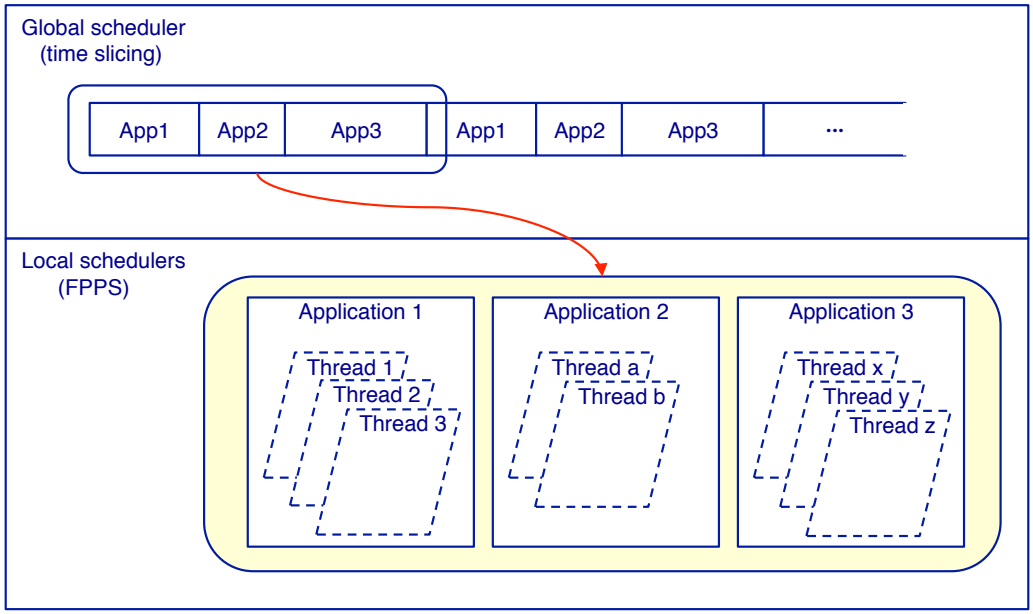

Figure 1. Hierarchical scheduling architecture.

tions available for the aeronautics domain, and the standard has been used in a number of commercial and military avionics systems. However, there is currently no open source platform available which can be used to build space partitioned systems in Ada. This abstract describes an open-source platform ORK+/XtratuM that follows the IMA approach. The ORK+/XtratuM platform has been built by combining the XtratuM hypervisor [8] with he Ada 2005 version of the Open Ravenscar Kernel (ORK+) [5, 12]. The prototype implementation of ORK+/XtratuM is targeted to LEON2 [7] and there are plans to make a complete implementation of ORK+/XtratuM to LEON3 [10] with MMU.

2. OVERVIEW OF ORK+/XTRATUM

XtratuM [8] is an open-source bare machine hypervisor that uses para-virtualization, i.e. the virtual machine interface is similar, but not identical, to the underlying hardware. Partitions are executed in processor user mode, whereas the hypervisor is executed in privileged processor mode giving a safe partition execution environment. In this way, the virtual machine interface gives access to the system resources through a set of system calls (*hypercalls*). As a result, an operating system that executes in an XtratuM partition has to be *para-virtualized*, which implies that some parts of the operating system hardware abstraction layer (HAL) have to be replaced with the corresponding hypercalls. Partitions are scheduled by XtratuM following a static cyclic plan which is defined by the system integrator. XtratuM version 2.2 is currently being used by CNES (Centre National d'Études Spatiales, France) as a time and space partitioning (TSP) based solution for building highly generic and reusable on-board payload software for space applications [1, 2].

ORK+ is an evolved version of ORK[4] that provides restricted tasking support as defined by the Ada Ravenscar profile. The kernel has been designed for efficient support of Ada tasking constructs, but can also be used with C programs, using a C interface package that is provided for this purpose. The current version, ORK+, includes support for the new Ada 2005 timing features which were included to support the timely execution of software component as required by the ASSERT Virtual Machine [13, 9].

The kernel lowest level components had to be re-implemented in order to port ORK+ to the virtual processor interface provided by the XtratuM hypervisor. As XtratuM uses para-virtualization, the processor-dependent components of ORK+ had to be re-written to access the processor resources by means of the XtratuM hypercalls. These lowest level components mainly deal with: CPU management, interrupt support, and time services. The resulting ORK+/XtratuM architecture is shown in figure 2. The figure shows several partitions based on ORK+/XtratuM, and one additional partition based on a bare machine C code running directly on top of XtratuM.

3. ADA EXTENSIONS FOR PARTITIONED SYSTEMS.

Additional services are requested by the the ARINC-653 standard to deal with a partitioned system, therefore a set of packages have to be defined to provide:

Memory Management Unit.

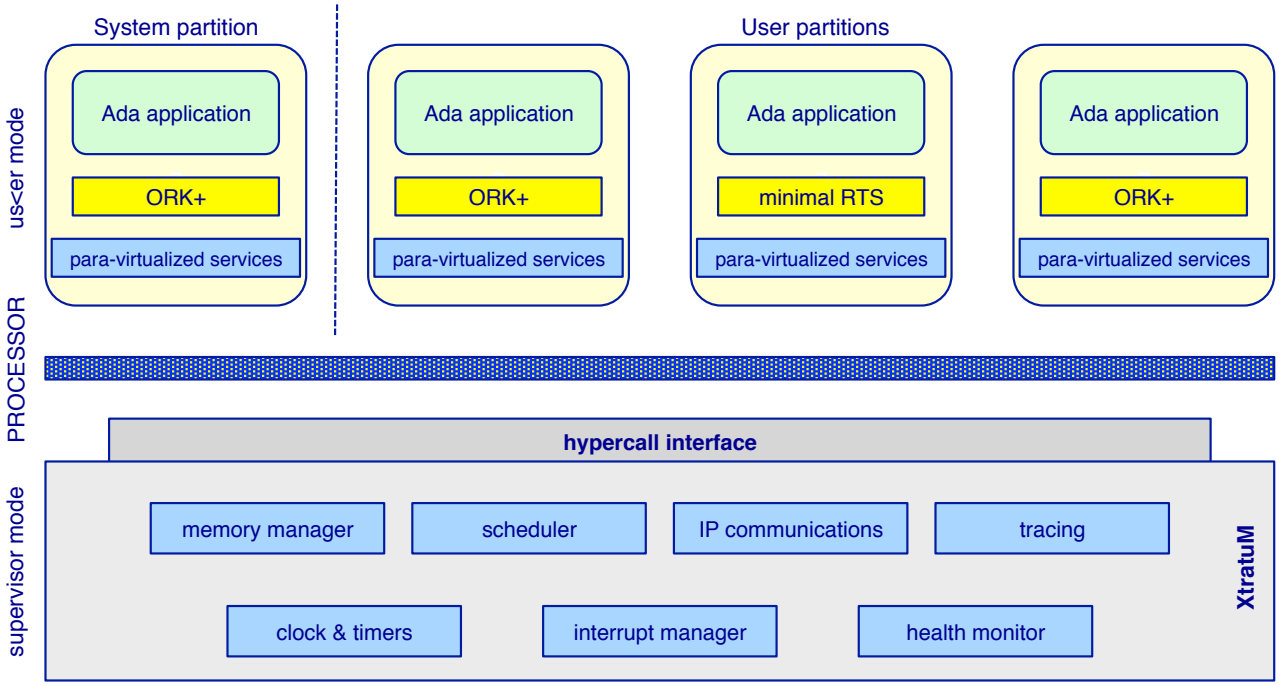

Figure 2. XtratuM architecture.

- Partition management: a system partition can observe the state of partitions or perform some actions (restart, stop, suspend, resume) on others.
- Inter-partition communication: partitions can communicate using sampling or queuing ports as defined in ARINC-653.
- Schedule plan management: a system partition can change the schedule plan.

A detailed description of these services and their implementation will be described in the final paper. Their specifications will follow the corresponding one that is written in the APEX specification grammar provided in the ARINC-653 standard.

CONCLUSION

The work presented in this abstract is a prototype implementation of the ORK+/XtratuM platform for LEON2 processors. There plans to port the platform to LEON3 processors with MMU support that could be available for final paper. LEON2 has little support to implement spatial isolation as it only has a set of fence registers and it is not possible to provide any protection against incorrect read operation. Moreover, this primitive support leads to a rigid memory sharing scheme between different partitions. These limitations will be overcome with the ORK+/XtratuM for LEON3 as LEON3 processors have a full-featured MMU.

Preliminary measures of performance gave promising results [6] demonstrating the advantage of a streamlined Ravenscar implementation over a para-virtualization hypervisor. The performance losses due to partition kernel layer were negligible with task periods longer than 10 ms, and only reached a significant value for a timer period of 1 ms.

REFERENCES

[1] P. Arberet, J.-J. Metge, O. Gras, and A. Crespo. TSP-based generic payload on-board software. In *DASIA 2009. DAta Systems In Aerospace.*, May. Istanbul 2009.

[2] P. Arberet and J. Miro. IMA for space : status and considerations. In *ERTS 2008. Embedded Real-Time Software.*, Jannuary. Toulouse. France 2008.

[3] ARINC. *Avionics Application Software Standard Interface — ARINC Specification 653-1*, October 2003.

[4] Juan A. de la Puente, José F. Ruiz, and Juan Zamorano. An open Ravenscar real-time kernel for GNAT. In Hubert B. Keller and Erhard Plödereder, editors, *Reliable Software Technologies — Ada-Europe 2000*, number 1845 in LNCS, pages 5–15. Springer-Verlag, 2000.

[5] Juan A. de la Puente, José F. Ruiz, Juan Zamorano, Rodrigo García, and Ramón Fernández-Marina. ORK: An open source real-time kernel for on-board software systems. In *DASIA 2000 — Data Systems in Aerospace*, Montreal, Canada, May 2000.

[6] Angel Esquinas, Juan Zamorano, Juan A. de la Puente, Miguel Masmano, Ismael Ripoll, and Alfons Crespo. ORK+/XtratuM: An open partitioning platform for Ada. In Alexander Romanovsky and Tullio Vardanega, editors, *Reliable Software Technologies — Ada-Europe 2011*, number 6652 in LNCS, pages 160–173. Springer-Verlag, 2011.

[7] Gaisler Research. *LEON2 Processor User's Manual*, 2005.

[8] M. Masmano, I. Ripoll, A. Crespo, J.J. Metge, and P. Arberet. Xtratum: An open source hypervisor for TSP embedded systems in aerospace. In *DASIA 2009. DAta Systems In Aerospace.*, May. Istanbul 2009.

[9] Enrico Mezzetti, Marco M.Panunzio, and Tullio-Vardanega. Preservation of timing properties with the Ada Ravenscar profile. In Jorge Real and Tullio Vardanega, editors, *Reliable Software Technologies — Ada-Europe 2010*, number 6106 in LNCS, pages 153–166. Springer-Verlag, 2010.

[10] Gaisler Research. Leon3 - high-performance sparc v8 32-bit processor. grlib ip core user's manual. http://www.gaisler.com.

[11] John Rushby. Partitioning for safety and security: Requirements, mechanisms, and assurance. NASA Contractor Report CR-1999-209347, NASA Langley Research Center, June 1999. Also to be issued by the FAA.

[12] Santiago Urueña, José Antonio Pulido, José Redondo, and Juan Zamorano. Implementing the new Ada 2005 real-time features on a bare board kernel. *Ada Letters*, XXVII(2):61–66, August 2007. Proceedings of the 13th International Real-Time Ada Workshop (IRTAW 2007).

[13] Juan Zamorano, Juan Antonio de la Puente, José Antonio Pulido, and Santiago Urueña. The ASSERT virtual machine kernel: Support for preservation of temporal properties. In *Data Systems in Aerospace — DASIA 2008*, Palma de Mallorca, Spain, 2008.